

BlowLTD's BDT Payment API

BlowLTD's Backend Developer Test API

This API is the answer to BlowLTD's Backend Developer Test. It represents part of a payments system API.

Requirements

- API should be RESTFUL
- API should be able to
 - Fetch a payment resource
 - Create, update and delete a payment resource
 - List a collection of payment resources
 - Persist resource state (e.g. to a database)

More information: <https://www.blowltd.com/>

Contact Info: jnl.mendonca@gmail.com

Version: V1

All rights reserved

<http://apache.org/licenses/LICENSE-2.0.html>

Access

Methods

[[Jump to Models](#)]

Table of Contents

[Default](#)

- [GET /api/v1/payments](#)
- [DELETE /api/v1/payments/{id}](#)
- [GET /api/v1/payments/{id}](#)
- [PATCH /api/v1/payments/{id}](#)
- [POST /api/v1/payments](#)

Default

GET /api/v1/payments

[Up](#)

Returns a list of payments. ([apiV1PaymentsGet](#))

Returns a list of the payments in the database.

Query parameters

limit (optional)

Query Parameter – The maximum number of payments to return in the response

offset (optional)

Query Parameter – The number of payments to skip before returning the response

Return type

array[[Payment](#)]

Example data

```
[ {
  "id" : "4ee3a8d8-ca7b-4290-a52c-dd5b6165ec43",
  "organisation_id" : "743d5b63-8e6f-432e-a8fa-c5d8d2ee5fcb",
  "version" : 0,
  "attributes" : {
    "amount" : 123.45,
    "currency" : "GBP",
    "beneficiary_party_id" : "11998de5-2b6a-44ae-a281-98794c539ec7",
    "debtor_party_id" : "db8e6fe6-1dd0-4b32-a0cc-9a96085bc88b",
    "sponsor_party_id" : "013f1dfc-f1c3-4ed8-b9f3-2d8b78154f96",
    "payment_scheme_type" : "FPS",
    "payment_type" : "Credit",
    "scheme_payment_type" : "ImmediatePayment",
    "scheme_payment_sub_type" : "InternetBanking",
    "reference" : "Payment for Em's piano lessons",
    "numeric_reference" : 123456789,
    "payment_id" : 123456789012345,
    "payment_purpose" : "Paying for goods/services",
    "end_to_end_reference" : "Wil piano Jan",
    "processing_timestamp" : 1566998801878
  }
}, {
  "id" : "4ee3a8d8-ca7b-4290-a52c-dd5b6165ec43",
  "organisation_id" : "743d5b63-8e6f-432e-a8fa-c5d8d2ee5fcb",
  "version" : 0,
  "attributes" : {
    "amount" : 123.45,
    "currency" : "GBP",
    "beneficiary_party_id" : "11998de5-2b6a-44ae-a281-98794c539ec7",
    "debtor_party_id" : "db8e6fe6-1dd0-4b32-a0cc-9a96085bc88b",
    "sponsor_party_id" : "013f1dfc-f1c3-4ed8-b9f3-2d8b78154f96",
    "payment_scheme_type" : "FPS",
    "payment_type" : "Credit",
    "scheme_payment_type" : "ImmediatePayment",
    "scheme_payment_sub_type" : "InternetBanking",
    "reference" : "Payment for Em's piano lessons",
    "numeric_reference" : 123456789,
    "payment_id" : 123456789012345,
    "payment_purpose" : "Paying for goods/services",
    "end_to_end_reference" : "Wil piano Jan",
    "processing_timestamp" : 1566998801878
  }
} ]
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

A JSON array of payments.

500

A server error occurred while processing request.

DELETE /api/v1/payments/{id}

Up

Delete a payment. (apiV1PaymentsIdDelete)

Deletes a payment matching the provided ID, from the database, if it exists.

Path parameters

id (required)

Path Parameter — The payment ID

Responses

204

Payment was deleted.

404

No payment matching the provided ID exists in the database.

500

A server error occurred while processing request.

GET /api/v1/payments/{id}

Up

Returns a payment. (**apiV1PaymentsIdGet**)

Returns a single payment from the database, if it exists and matches the provided ID.

Path parameters

id (required)

Path Parameter — The payment ID

Return type

[Payment](#)

Example data

Content-Type: application/json

```
{
  "id" : "4ee3a8d8-ca7b-4290-a52c-dd5b6165ec43",
  "organisation_id" : "743d5b63-8e6f-432e-a8fa-c5d8d2ee5fcb",
  "version" : 0,
  "attributes" : {
    "amount" : 123.45,
    "currency" : "GBP",
    "beneficiary_party_id" : "11998de5-2b6a-44ae-a281-98794c539ec7",
    "debtor_party_id" : "db8e6fe6-1dd0-4b32-a0cc-9a96085bc88b",
    "sponsor_party_id" : "013f1dfc-f1c3-4ed8-b9f3-2d8b78154f96",
    "payment_scheme_type" : "FPS",
    "payment_type" : "Credit",
    "scheme_payment_type" : "ImmediatePayment",
    "scheme_payment_sub_type" : "InternetBanking",
    "reference" : "Payment for Em's piano lessons",
    "numeric_reference" : 123456789,
    "payment_id" : 123456789012345,
    "payment_purpose" : "Paying for goods/services",
    "end_to_end_reference" : "Wil piano Jan",
    "processing_timestamp" : 1566998801878
  }
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

A JSON detail of the payment. [Payment](#)

400

Payment retrieval failed. ID is not valid. [Error](#)

404

No payment matching the provided ID exists in the database.

500

A server error occurred while processing request.

PATCH /api/v1/payments/{id}

Up

Update a payment. ([apiV1PaymentsIdPatch](#))

Updates a payment matching the provided ID, in the database, if it exists.

Path parameters

id (required)

Path Parameter – The payment ID

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [PaymentInput](#) (required)

Body Parameter – The JSON detail of the payment

Return type

[Payment](#)

Example data

Content-Type: application/json

```
{
  "id" : "4ee3a8d8-ca7b-4290-a52c-dd5b6165ec43",
  "organisation_id" : "743d5b63-8e6f-432e-a8fa-c5d8d2ee5fcb",
  "version" : 0,
  "attributes" : {
    "amount" : 123.45,
    "currency" : "GBP",
    "beneficiary_party_id" : "11998de5-2b6a-44ae-a281-98794c539ec7",
    "debtor_party_id" : "db8e6fe6-1dd0-4b32-a0cc-9a96085bc88b",
    "sponsor_party_id" : "013f1dfc-f1c3-4ed8-b9f3-2d8b78154f96",
    "payment_scheme_type" : "FPS",
    "payment_type" : "Credit",
    "scheme_payment_type" : "ImmediatePayment",
    "scheme_payment_sub_type" : "InternetBanking",
    "reference" : "Payment for Em's piano lessons",
    "numeric_reference" : 123456789,
    "payment_id" : 123456789012345,
    "payment_purpose" : "Paying for goods/services",
    "end_to_end_reference" : "Wil piano Jan",
    "processing_timestamp" : 1566998801878
  }
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

A JSON detail of the payment. [Payment](#)

400

Payment update failed. Client input is not valid. [Error](#)

404

No payment matching the provided ID exists in the database.

500

A server error occurred while processing request.

[Up](#)

POST /api/v1/payments

Creates a payment. ([apiV1PaymentsPost](#))

Creates a payment with the parameters sent in the request's body. Returns the payment details if successful.

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [PaymentInput](#) (required)

Body Parameter – The JSON detail of the payment

Return type

[Payment](#)

Example data

Content-Type: application/json

```
{
  "id" : "4ee3a8d8-ca7b-4290-a52c-dd5b6165ec43",
  "organisation_id" : "743d5b63-8e6f-432e-a8fa-c5d8d2ee5fcb",
  "version" : 0,
  "attributes" : {
    "amount" : 123.45,
    "currency" : "GBP",
    "beneficiary_party_id" : "11998de5-2b6a-44ae-a281-98794c539ec7",
    "debtor_party_id" : "db8e6fe6-1dd0-4b32-a0cc-9a96085bc88b",
    "sponsor_party_id" : "013f1dfc-f1c3-4ed8-b9f3-2d8b78154f96",
    "payment_scheme_type" : "FPS",
    "payment_type" : "Credit",
    "scheme_payment_type" : "ImmediatePayment",
    "scheme_payment_sub_type" : "InternetBanking",
    "reference" : "Payment for Em's piano lessons",
    "numeric_reference" : 123456789,
    "payment_id" : 123456789012345,
    "payment_purpose" : "Paying for goods/services",
    "end_to_end_reference" : "Wil piano Jan",
    "processing_timestamp" : 1566998801878
  }
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

201

A JSON detail of the payment. The creation was successful. [Payment](#)

400

Payment creation failed. Client input is not valid. [Error](#)

500

A server error occurred while processing request.

Models

[[Jump to Methods](#)]

Table of Contents

1. [Currency](#)
2. [Error](#)
3. [Id](#)
4. [Payment](#)
5. [PaymentAttributes](#)
6. [PaymentAttributesInput](#)
7. [PaymentInput](#)
8. [PaymentSchemeType](#)
9. [PaymentType](#)
10. [SchemePaymentSubType](#)
11. [SchemePaymentType](#)

Currency

[Up](#)

Currency enumerator. Contains allowed values for currency type.

Error

[Up](#)

Error object. Used in error responses that require extra details.

message (optional)

[String](#)

example: A description of the error

type (optional)

[String](#)

example: ErrorTypeString

code (optional)

[String](#)

example: 123

Id

[Up](#)

Id string. Contains the pattern for allowed IDs (UUID v4).

Payment

[Up](#)

Payment object. Main API resource.

id (optional)

[Id](#)

version (optional)

[BigDecimal](#)

example: 1

organisation_id (optional)

[Id](#)

attributes (optional)

[PaymentAttributes](#)

PaymentAttributes

[Up](#)

Payment Attributes object. Contains most of the information related to a payment.

amount (optional)

[BigDecimal](#) Amount paid

example: 123.45

currency (optional)

[Currency](#)

beneficiary_party_id (optional)

[Id](#)

debtor_party_id (optional)

[Id](#)

sponsor_party_id (optional)

[Id](#)

payment_scheme_type (optional)

[PaymentSchemeType](#)

payment_type (optional)

[PaymentType](#)

scheme_payment_type (optional)

[SchemePaymentType](#)

scheme_payment_sub_type (optional)

[SchemePaymentSubType](#)

reference (optional)

[String](#)

example: A description of the payment

payment_purpose (optional)

[String](#)

example: The purpose of the payment

end_to_end_reference (optional)

[String](#)

example: Another description of the payment

processing_timestamp (optional)

[String](#) Timestamp for the payment's processing operation. Replaces the processing_date string.

example: {}

numeric_reference (optional)

[String](#)

example: 5349

payment_id (optional)

[String](#)

example: 1234512345

PaymentAttributesInput

[Up](#)

Payment Attributes object. Contains most of the information related to a payment.

amount (optional)

[BigDecimal](#) Amount paid

example: 123.45

currency (optional)

[Currency](#)

beneficiary_party_id (optional)

[Id](#)

debtor_party_id (optional)

[Id](#)

sponsor_party_id (optional)

[Id](#)

payment_scheme_type (optional)

[PaymentSchemeType](#)

payment_type (optional)

[PaymentType](#)

scheme_payment_type (optional)

[SchemePaymentType](#)

scheme_payment_sub_type (optional)

[SchemePaymentSubType](#)

reference (optional)

[String](#)

example: A description of the payment

payment_purpose (optional)

[String](#)

example: The purpose of the payment

end_to_end_reference (optional)

[String](#)

example: Another description of the payment

numeric_reference (optional)

[String](#)

example: 5349

payment_id (optional)

[String](#)

example: 1234512345

PaymentInput

[Up](#)

Payment object. Main API resource.

version (optional)

[BigDecimal](#)

example: 1

organisation_id (optional)

[Id](#)

attributes (optional)

[PaymentAttributes](#)

PaymentSchemeType

[Up](#)

Payment Scheme Type enumerator.

PaymentType

[Up](#)

Payment Type enumerator.

SchemePaymentSubType

[Up](#)

Scheme Payment Sub-Type enumerator.

SchemePaymentType

[Up](#)

Scheme Payment Type enumerator.